- **Code Reflection:** A brief explanation of the code and its purpose, and a brief discussion of your experience in developing it, including any issues that you encountered while completing the exercise and what approaches you took to solve them.
    - This code should create a linked list of the Monthly Bid data. It will add, delete, display all bids, and search for a specific bid. This one wasn't too difficult to put together. The hardest part for me was understanding exactly how to make it work. Once I figured that out, it was just a matter of figuring out the syntax to make it all work.

- **Pseudocode or Flowchart:** A pseudocode or flowchart description of the code that is clear and understandable and captures accurate logic to translate to the programming language.

```
START main()

INITIALIZE variables

DISPLAY Menu

    GET user_input

        IF "1":

            CALL getBid():

                GET Id, Title, Fund, and Amount

                CREATE "Bid" object

                RETURN "Bid"

            BREAK

        IF "2":

            START clock

            CALL loadBids():

                PRINT "Loading CSV File"

                FOR EACH row in the csv file:

                    GET the Id, Title, fund, and amount

                    CREATE "Bid" object

                    APPEND to list
```

```
        END clock
        PRINT clock times in seconds
        BREAK
IF “3”:
        CALL PrintList():
                CREATE “current” NODE
                WHILE “current”:
                        PRINT bidID: Title, Amount, Fund
        BREAK
IF “4”:
        START clock
        CALL Search():
                INIT “search” NODE = head
                IF bidId = “search” bidId:
                        Head = the node after “search”
                        DECREASE size of list by 1
                        RETURN
                WHILE “search”:
                        IF bidId = “search” bidId:
                                RETURN “search” bid
                        ELSE:
                                SET “Search” AS the NODE after “search
                RETURN Bid()
        END clock
```

```
        IF bid is NOT EMPTY:
            CALL displayBid()
        ELSE:
            PRINT "Bid Id _____ not found"
        PRINT clock times in seconds
        BREAK
IF "5":
        CALL remove():
            INIT "oldHead" NODE AS head
            IF "oldHead" = nullptr :
                RETURN
            IF "oldHead" bidID = bidId:
                Head = NODE after "oldHead"
                DELETE "oldHead"
                DECREASE size of list by 1
                RETURN
            INIT "toBeDeleted" as head
            WHILE "toBeDeleted" has something after it:
                IF "bidID" = The bidID after "toBeDeleted":
                    INIT "temp" NODE as the NODE after "toBeDeleted"
                    the NODE after "toBeDeleted" = 2 NODEs after "toBeDeleted"
                    DELETE "temp"
                    DECREASE size of list by 1
                RETURN
```

```
            "toBeDeleted" = the NODE after "toBeDeleted"

        BREAK


PRINT "Goodbye"
```